Resource Pricing for Differentiated Services

Peter B. Key

Microsoft Research, St George House, Cambridge, CB2 3HN, UK
http://research.microsoft.com/users/pbk/

Abstract. In paper we present an overview of recent work on resource pricing for differentiated services in the Internet. This approach is based upon encouraging cooperation between the end-systems and the network by use of the correct feedback signals. These signals reflect the congestion shadow prices at a resource, and their use means then even 'selfish' end-systems, acting in their own best interests, will push the system to a global or social optimum. In contrast to most current Diffserv proposals, little is required from resources in the network; they just have to mark packets correctly, while the end-system can use complex or simple strategies. All that is needed is for the end-systems to have an incentive to react to the feedback signals, and then we have a distributed resource sharing mechanism. We give examples of typical end-system behaviour, and show how this approach can also implement Distributed Admission Control, where the decision is in the hand of the end-system. We comment on how ECN (Explicit Congestion Notification) could be used as an enabling technology. Lastly we outline how guarantees can be constructed with this framework.

1 Introduction

The current Internet is based upon a single class of service (Best Effort), with a limited range of end-system behaviour. But different applications, particularly real-time services, may have different requirements from the network, in other words demand a different Quality of Service, (QoS), and the IETF is seeking to evolve the Internet to meet these demands. Quality of Service has layers of meaning, but at its most basic says something about the level and quality of bandwidth delivered, and the delay seen by packets traversing the Network. Telephony networks assure quality of service by dimensioning for a peak load, using signalling, reserving bandwidth for connections and denying access to connections if the network is congested. The IntServ proposal for the Internet incorporates some of these ideas to give reservations some assured bandwidth. Current DiffServ proposals look at ways of giving differentiated services by defining marking packets with various labels (code points), treating packets differently on a per-hop basis and using service level agreements to give some form of end-to-end performance guarantees across domains. Traffic shapers and policers are needed at the edges and gateways to enforce contracts, with priority queueing or some other form of scheduling needed in the routers to produce the different per-hop behaviours.

There is a growing body of work that looks at a simpler framework for providing differential QoS, using ideas from economics and optimisation based upon 'Resource Pricing' or 'Congestion Pricing'. This is the approach we explore in this paper. The basic idea is to encourage co-operation between the users and the network by sending the correct signals to the users. These reflect the damage that the users cause to the network by entering the system, or the 'congestion costs'. For example, if the network is lightly loaded, these are negligible, but increase with the network loading. A network comprises a set of linked resources, each of which calculate their own 'Resource Price' or 'Congestion Cost', which are aggregated across routes by users. The users (end-systems or applications) are then free to act as they wish, subject to the constraint that there is an incentive to react to feedback signals. In economic terms congestion is a (negative) externality, and if this information is fed back to the users as a cost or tax, then the externality is internalised and users acting in their own self-interest will push the system to the global or system optimum.

This framework enables a distributed resource sharing approach to QoS, where the end-systems play a key role in the resource allocation, and cooperate with each other and the network through the feedback signals that the network provides. It can also be seen as a practical way of implementing a version of the 'smart-market' approach proposed by Mackie-Mason and Varian [15]. Much less is required of the network than current DiffServ proposals, and more of the intelligence is placed in the end-systems or edge devices. An interesting insight by Bonald and Massoulié [2] suggest that the use of priority mechanisms, such as those used in Diffserv may cause instability in the network, whereas the feedback signals we advocate do not suffer from this drawback.

In section 2 we explore the theoretical background to this work, which is based of the work on Kelly and co-workers [7, 4]. Section 3 uses a simple model of TCP to relate the theory to current practice, while Section 4 briefly considers implementation issues and what is required of routers. It should be stressed that in this paper we do not address the issue of how to charge for services: resource pricing only reflects congestion costs, which may or may not be translated into real monetary costs to the user, and other considerations have to be taken into account when setting tariffs or prices. Our approach can be thought of as a distributed game between the users and the network, which we look at in Section 5, while Section 6 looks at end system behaviour, for adaptive applications and file transfers. It is also possible to deal with demands that have a fixed bandwidth requirement, putting admission control in the hands of the user, which we describe in Section 7. Lastly we comment on the relationship between guarantees and this approach, and make some concluding remarks.

2 Motivation and Theoretical Background

We now explain how the feedback signals can be derived, and how they link to notions of 'fairness'. For convenience in what follows we use the term 'user' to denote any end-system or application.

A network is a set of linked resources, and a path through the network will use some subset of these resources in a specific order. The description is general, but it may help to think of the current Internet where typical resources are bandwidth and service rate or buffer capacity in a router output port. We shall assume fixed routing, so a route determines a specific subset of the resources. Let J be a finite set of resources indexed by j, and R a finite set of users (end-systems), indexed by r, where the 0–1 incidence matrix $A = A_{jr}$ indicates whether user r uses resource j or not. Suppose further that we can characterise a user's preference for bandwidth by a concave, non-decreasing utility function, $U_r(x)$, which is appropriate for 'elastic' traffic [22]. For convenience assume U_r is everywhere differentiable and $U_r(x_r) \downarrow -\infty$ as $x \downarrow 0$. It is natural for users to attempt to pursue their own ends, and seek to maximise $U_r(x_r)$ over $x_r \geq 0$. But resources are (usually!) finite, so this behaviour will tend to overload certain resources. To counteract this, suppose that when the load on resource j is y_j, the resource incurs a cost at rate $C_j(y_j)$. Then a social planner would seek to

$$\text{Maximise} \quad \sum_r U_r(x_r) - \sum_j C_j\left(\sum_r A_{jr}x_r\right) \qquad \text{over} \quad x_r \geq 0 \;; \quad r \in R \;. \tag{1}$$

At the optimum,

$$U_r'(x_r) = \sum_{j \in r} p_j(y_j) \tag{2}$$

where p_j is the derivative or shadow price $p_j(y_j) = C_j'(y_j)$ with a corresponding load on resource j given by

$$y_j = \sum_r A_{jr}x_r \;. \tag{3}$$

Suppose now we send feedback signals to the user. We want signals to be additive, and if signals are to be carried on packets, signals need to be proportional to load, suggesting a feedback signal of the form $t_r x_r$. If these reflect a 'charge' to the User, the user wants to maximise the net return, that is

$$\text{Maximise } U_r(x_r) - t_r x_r \qquad \text{over} \quad x_r \geq 0. \tag{4}$$

hence if these 'prices' are set correctly, that is if

$$t_r = \sum_{j \in r} p_j(y_j) \tag{5}$$

the users acting independently will drive the system towards the social optimum [7]. Note that the tax or congestion price t_r internalises the congestion costs, and is sum of the resource prices along the route.

Under this identification, and writing $p_r = \sum_{j \in r} p_j(y_j)$, a natural user adaptation is to adapt the rate x_r according to

$$\frac{d}{dt}x_r(t) = \kappa_r\left(x_r(t)\,U_r'(x_r(t)) - x_r(t)\sum_{j \in r} p_j(y_j(t))\right) \tag{6}$$

in which case provided U_r is strictly concave and each C_j is convex then each user will converge to the unique (system). It is possible to show by Lyapunov techniques that all trajectories converge to the unique fixed point [7, 10].

We now show that is it possible for the Network to mandate all users to use a particular update rule (perhaps by mandating a specific protocol stack), which will impose a particular form of fairness, and still recover the same results. Part of the motivation for this is that, in general, the network does not know the utility functions of the users (which are an abstraction in any case). Suppose that each user has to use the 'willingness-to-pay' update rule

$$\frac{d}{dt}x_r(t) = \kappa_r \left(w_r - x_r\left(t\right) \sum_{j \in r} p_j\left(y_j\left(t\right)\right) \right) \quad . \tag{7}$$

This is equivalent to using a utility function of the form

$$F_r(x) = w_r \log x_r \tag{8}$$

and hence the Network is implicitly maximising $\sum_r w_r \log x_r - \sum_j C_j(y_j)$. It can be shown that such an allocation is weighted Proportionally Fair [7]. [1] At the optimum,

$$w_r = p_r x_r \tag{9}$$

hence w_r is the amount user r is prepared to pay per unit time. If the user adapts the parameter w_r over time according to

$$w_r = x_r U_r'\left(x_r\right) \tag{10}$$

then the User Optimum and Social optimum again coincide, based upon an underlying 'Proportional Fairness' model. Updating w_r according to (10) is equivalent to the user seeking to

$$\text{Maximise} \quad U_r\left(\frac{w_r}{p_r}\right) - w_r \quad \text{over} \quad w_r \geq 0 \ . \tag{11}$$

We shown in the appendix that the same results hold under different mandated behaviours dictated by choosing a different function F. But much of the literature has concentrated on the 'willingness to pay' algorithm, which is equivalent to $F_r(x_r) = w_r \log x_r$. This which is appealing since firstly, w_r has a ready interpretation, secondly the behaviour of aggregates is linear with respect to w_r (in other words a single stream with value $w = w_1 + w_2$ behaves as the sum of the two streams with w_1 and w_2) and thirdly the underlying fairness model, weighted 'Proportional Fairness' [7], in economic terms is 'Nash fair' and results in an Nash arbitration scheme [20, 18]. In the bandwidth sharing case it is possible to show that this is only allocation that produces a Nash arbitration scheme.

3 TCP as an Optimisation

The above description laced with talk of utility functions may seem very abstract. However, consider the current TCP protocol. When in congestion avoidance mode, the congestion window W increases by 1 per round trip time, and halves its window if a packet loss is detected. If the same behaviour occurs when a packet is marked (rather than lost), then in a window of size W, if packets are marked with probability p, there will be pW packet marks each potentially halving the window. Now scale the systems so that W is large, then for small p the rate of packet sending $x = W/T$, where T is the (assumed constant) round trip time (RTT) can approximated by

$$\frac{dx(t)}{dt} = \frac{1}{T}\frac{dW(t)}{dt} = \frac{1}{T^2} - \frac{x(t)^2 p}{2} \tag{12}$$

At the equilibrium this gives the familiar inverse square root dependence on p

$$x = \frac{1}{T}\sqrt{\frac{2}{p}} \ . \tag{13}$$

[1] Normally fairness is used in the context of fixed capacity constraints, where resource j has capacity c_j; this fits in the above by taking $C_j(\cdot)$ as a penalty function.

This is the same *as if* each user was trying to maximise their *net* utility, that is utility minus cost, where the utility function is

$$U(x) = K - \frac{2}{T^2 x} \tag{14}$$

for K an arbitrary constant, and where the cost function is just the linear rate of charge, px. This utility function can be interpreted as a weighted version of the time taken to transfer a file of unit size [11]. Note how this utility function penalises long round-trip times (weights round trip times more heavily than bandwidth), and that squared dependence on x in the decrease behaviour means that the behaviour of aggregates is not linear.

4 Implementation Issues and Marking

4.1 End-system Reactions

To implement the resource pricing ideas, a way of generating feedback signals is needed, and users (end-systems or applications) must be able to adapt. Resources generate feedback signals, and we explore various ways of setting these marks below. If the feedback signals can be represented by a single bit, then the current IETF Explicit Congestion Notification (ECN) RFC [21] provides a suitable mechanism: this sets a flag in the IP header, which can be used to indicate congestion or not. The current RFC concentrates on TCP behaviour, and specifies how TCP should behave in response to marked packets, by essentially requiring the same behaviour as if the packet had been lost. By allowing a much more general behaviour, we lay the framework for truly differential quality of service. For responsive flow-aware applications, such as TCP, the feedback signals can be fed back to the receiver in the ACK packet (as in the ECN RFC). For other applications (such as current UDP-type applications) the feedback needs to be returned to the source. One approach that enables existing applications to be used is to create some form of 'Congestion Manager', which sits between the application and the network stack and interprets the feedback signals to the application, together with a mechanism for reflecting the marks back to the source. The reflection may either be done on a per-resource basis, or reflected by an edge device. Prototype operating systems stacks exist which support ECN, and ECN-capability is likely to be in future versions of Operating Systems such as Windows and Linux, thus laying the foundations for building new adaptive applications, which can react directly to the feedback information without the need for a Congestion Manager.

It is an open question as to whether single bit marking is sufficient: if more bits are required then the ECN proposal is not sufficient, and either IPv6 would have to be used for a general solution, or the Congestion manager approach for IPv4.

4.2 Resource Marking

The question of how to mark packets depends on the cost function $C(y)$ that is used. Current marking in the Internet is based around marking lost packets, where the marking function, $p(y)$ (the derivative of $C(y)$) represents the rate of loss. But it is more natural to regard the *cost* function $C(y)$ as reflecting the rate of loss; for example if a resource has capacity c_j then the cost should reflect the rate of loss, $(y_j - c_j)^+$, or $\mathbb{E}\left[(y_j - c_j)\right]$ if we interpret y_j as a random variable, If y_j is Poisson [4] or Gaussian, then the derivative is given by

$$p_j(y_j) = \Pr\left\{y_j \geq c_j\right\} \equiv \mathbb{E}\left[1_{y_j \geq c_j}\right] \quad , \tag{15}$$

in other words we mark all the packets when the load exceeds capacity. This is the probability of resource saturation, which typically marks at least an order of magnitude more packets than the loss rate (which is $\mathbb{E}\left[(y - c_j)^+\right] / \mathbb{E}[y]$). Note the difference here: loss marking marks those packets which suffer as a result of congestion, whereas the alternative approach advocated here marks all those packets *responsible* for congestion.

Current generation routers are reasonably approximated by an output buffered switch, with packets dropped (marked) at an output port when the buffer exceeds its maximum value B. Recent proposals such as RED (Random Early Detection) [3], start marking packets before loss occurs, when some threshold $b < B$ is met, and try to avoid synchronisation effects that occur when loss is used as a feedback signal by marking some proportion of packets at levels below b. A related proposal (REM, Random Early Marking) [13] marks a packet with probability $1 - \phi^{-W}$) where W is a congestion measure based on current workload and $\phi > 0$ a parameter.

The choice of an appropriate $p(\cdot)$ is the subject of current research, see for example [25]. The beginning of this section suggested marking on arrival rate, which is a more reactive signal than marking schemes based on queue

length, since the queueing process integrates the arrival rate. A virtual queue marking scheme [4] which marks as though the queue had a smaller service rate (capacity) and potentially smaller buffer size is able to provide early warning of problems. Moreover by a suitable choice of parameters [6], we can track the derivate of the real queue, which is related to the arrival rate, and the real object of interest.

More generally, the function $C(y)$ might represent the costs associated with delay, in which case the price functions p reflect the shadow price of delay.

5 Distributed Games

With the network providing the correct feedback signals (the shadow prices), users or end-systems can do as they please, provided that they have an incentive to react to these prices or 'costs'. The signals encourage cooperation, while the users seek to do the best they can subject to the cost they incur. We can then view the whole system as an environment, where users 'compete' against each other. This is reminiscent of a multi-user, multi-objective game where users may have very different objectives, and seek the best way of achieving them. For example, one user may wish to maximise throughput whilst ensuring the rate of charging, the cost, is less than some amount, while some users may be prepared to adapt the rate in response to price signals, and another may not.

The 'best' algorithm for a user will be one that performs well in a mixed environment: the strategy has to do well against all sorts of other strategies. Consequently, one way to discover 'good' strategies is to create an environment where strategies can compete against each other. A first step is to construct strategies that do well for a specific object, and then see how robust they are. A good algorithm can then be embedded in a protocol, which adapts the rate of sending packets in response to feedback received from the network.

Microsoft Research in Cambridge has built such an environment [9, 10], where such ideas can be tried out: a network simulator simulates certain key features of a real network (transmission of packets, routers, etc) and a simple text-based protocol allows users to communicate with the network. This enables users to write strategies in any language, and communicate with the network remotely via a TCP connection. In other words this is a distributed game environment, where users 'compete' against each other and the network. The network topology, router functionality and marking functions can be freely altered to allow general experiments to be conducted.

In the next section we describe certain user strategies.

6 End-system behaviour

The simplest form of user behaviour is an unreactive source, which just continues at a fixed or varying rate, regardless of the feedback. These can be interpreted as users who are insensitive to price, and have the effect of forcing the price up (price setters). We now look at those adaptive applications that adapt their sending rate in response to feedback signals, which can be thought of as generalisations of TCP.

6.1 Rate Control

Suppose users adjust their rate according to the Gibbens and Kelly 'willingness-to-pay' strategy [4] described earlier,

$$\frac{d}{dt}x_r(t) = \kappa_r \left(w_r - p_r(t)x_r(t)\right) \tag{16}$$

where as in Section 2, $p_r(t)$ is the feedback along the route, the sum of the resource prices $p_r(t) = \sum_{j\in r} p_j(t)$. Then w_r is the amount user r is prepared to pay per unit time, or the maximum rate of marking the user can tolerate, and κ_r is a gain parameter. The user increases the sending rate at rate κ_r times the difference between what the user is prepared to pay (w_r) and the network charges $p_r x_r$. In the steady state equilibrium, users have a throughput proportional to w_r, i.e. $x_r = w_r/p_r$, hence those that are prepared to pay twice as much receive twice as much. Note the dependence on $1/p_r$, in contrast to the square root dependence of TCP. Simulation studies [4, 10, 12] have shown that this relative throughput is indeed attained in practice and also show [10, 12]how such strategies can co-exist with TCP [10]. Therefore this simple mechanism has given a way of giving relative shares. Recall that this algorithm is equivalent to using a logarithmic utility function, $U_r(x_r) = w_r \log x_r$ hence the resulting allocations are weighted proportionally fair. This algorithm with a packet scheduling policy that conforms to the rate equation can define a rate control protocol.

The parameter κ_r affects the rate of convergence. Equation (16) always converges to the equilibrium point, however this assumes instantaneous feedback. In practice, there is a delay D_{sj} from source s to resource j, and a delay in the message travelling from resource j back to the sender, D_{js}. Let T_r be the round-trip time for user r, and assume that $D_{rj} + D_{jr} = T_r$ then the delayed version of equation (16) is

$$\frac{d}{dt}x_r(t) = \kappa_r \left(w_r - p_r(t - T_r)x_r(t - T_r)\right) \tag{17}$$

where now $p_r(t - T_r)$ is shorthand notation for $\sum_{j \in r} p_j(\sum_{s:j \in s} x_s(t - D_{jr} - D_{sj})$. Recent work by Massoulié [16] has proved a version Tan's conjecture [24] that this system is stable if

$$\kappa_r T_r \left(p_r + \sum_{j \in r} p'_j \bar{y}_j \right) < 1 \ , \quad r \in R \tag{18}$$

where $\bar{y}$ the equilibrium vector of loads. This says that κ_r must be less than some fraction of the inverse round trip time, $1/T_r$. Note that the multiplier only depends on quantities along route r. The larger the κ_r, the faster the convergence, but if κ exceeds the bound the system may be unstable. Note that the system can still oscillate — the right hand side of (18) has to be smaller than 1 to avoid oscillation, and in the single resource case less than $1/e$.

The above evolutions are gradual adaptations designed to adjust the load to its equilibrium point. When a connection first enters, it can take some time to reach this point, and there are good reasons to use something like TCP slow-start behaviour to find a good operating point: for example doubling the sending rate until say w_r packets are marked, and then using the slow evolution. Equivalently, use the parameter κ_r as a time dependent parameter, $\kappa_r = \kappa_r(t)$ with κ_r starting out from a high (unstable) value and decaying to a stable value. [8] give an interesting description of slow-start like algorithms in terms of risk-averse behaviour.

6.2 Window Based Control

Current TCP is a window based control, which has useful self-clocking features. Writing $x_r = W_r T_r$, and $\kappa_r = \gamma_r / T_r$ gives the window evolution for W_r as

$$\frac{d}{dt}w_r(t) = \gamma_r \left(w_r - \frac{p_r(t)}{T_r} W_r(t) \right) \tag{19}$$

with γ_r a suitable fraction to give stability. This can be implemented by increasing the window $W_r(t)$ by $\gamma_r w_r T_r$ every round trip time (equivalently increasing the window by $\gamma_r w_r T_r / W_r$ every ACK), and decreasing by γ_r for every marked ACK. This gives an expected throughput which is independent of the round trip time: if an inverse dependence on the round trip time is required (to be TCP like) then the updating can be modified to

$$\frac{d}{dt}w_r(t) = \gamma_r \left(\frac{w_r}{T_r} - \frac{p_r(t)}{T_r} W_r(t) \right) \tag{20}$$

where now the window increases by $\gamma_r w_r / W_r$ every ACK, and where for stability $\gamma_r < 1/\left(p_r + \sum_{j \in r} p'_j \bar{y}_j\right)$. This is closest to a direct modification of TCP: the throughput is given by

$$\bar{x}_r = \frac{1}{T_r} \frac{w_r}{p_r} \ . \tag{21}$$

6.3 File Transfers

File transfers have rather different objectives: example objectives might be to transfer a file F by a given time T at minimum cost, or transfer a file F at a cost of no more than W. These have very different behaviour: in the first case, [5] used the simulation environment to look at simple strategies, where the sending rate could vary between a high peak rate (sending packets close together) or a low rate. Simple strategies, such as sending at the high rate if the last sent packet was not marked performed well in bake-off experiments. This is not surprising, and echoes results found by Axelrod [1] for the repeated prisoner's dilemma where simple strategies proved the

most robust in a mixed environment (despite not being 'optimal' when considered in isolation). Massoulié and Key [17] looked at a more sophisticated estimation procedure that could be used to try and estimate marking periods and react accordingly.

If instead the user wants to send the file size F at a cost of no more than W, then users are prepared to pay a maximum amount per packet, $W(t)/F(t)$, where $W(t)$ denotes the amount of W remaining at time t and $F(t)$ the amount still to be transferred. Hence a user will enter the system if the price is below this, and will drop out if the predicted price means that they cannot afford to send the amount left. This encourages a start-stop behaviour where a sender can stop sending in the middle of a file, and wait until the price drops again before re-entering. This can be fitted into a 'willingness to pay' strategy [4], where the willingness to pay, $w(t)$, is a function of time and updated by relating it to $W(t)x(t)/F(t)$.

For a general user, whose utility function is a decreasing function of the time to transfer T, with a maximum bearable price $p^* = W/F$, Key and Massoulié [8] show that for a large system, where users and capacity are scaled together, the optimal strategy for a user is to send at the peak rate, if the price is less than p^* and otherwise to wait. This provides some linkage between the previous two strategies.

7 Distributed Admission Control

So far we have concentrated on concave utility functions. If the utility function is 's-shaped', with a convex initial region, followed a concave region, then if the price is too high, the user will not enter (this happens if px never intersects the curve $U(x)$), otherwise the user will ideally choose the point rate x where $U'(x) = p$, and the convex initial region has no influence on the allocation. Let us now concentrate on a users who only want to enter if the price is less than some value p^*, and want to have fixed amount of bandwidth, x_f (f for fixed) . This could correspond to users who have a utility function which is approximates to a step function, appropriate for non-adaptive real time traffic. Now suppose each connection sends a number of probe packets through the network, and only enters if none of these packets are marked. This creates a distributed admission scheme studied by Kelly et al [6], where the users, rather than the network, decide if it they should enter or not.

Let m_f be the number of probe packets user f send through the network. The larger m_f, the less likely user r will enter: in effect the user is trading off the cost of entering the network and being marked more than they want to be against not entering and losing utility. Key and Massoulié [8] suggest a Bayesian framework for choosing m_f. Suppse we have a set of such users, F, that connections of type f arrive as a Poisson process of rate ν_f, last for a mean time μ_f and when connected generate packets at rate λ_r when connected. If mean holding times and packet generation rates are equal, and if this set of users are the only users of a network or sub-network, then the distribution of the number of connections n_f has a product form solution. As we relax these assumptions, the product form disappears but we can write down differential equations that approximate the system behaviour, and which become exact as the system size grows.

For example, suppose we now mix the adaptive and non-adaptive traffic, and that instances of adaptive traffic of type r arrives with Poisson rate ν_r and have a mean holding time of μ_r. Then the evolution of the system is described by the system of equations,

$$\dot{n}_r(t) = \nu_r - n_r(t)\mu_r \tag{22}$$

$$\dot{x}_r(t) = \kappa_r \left(w_r - p_r(t)x_r(t)\right) \tag{23}$$

$$\dot{n}_f(t) = \nu_f(1 - p_f(t))^{m_f} - n_f(t)\mu_f \tag{24}$$

where

$$p_r(t) = \sum_{j \in r} p_j(y_j(t)) \tag{25}$$

$$y_j(t) = \sum_{r \ni j} n_r(t)x_r(t) + \sum_{f \ni j} n_f(t)\lambda_f \tag{26}$$

The last equation implies that when a new type r connection arrives it uses the common sending rate of the type r connections.

We can easily write down a Lyapunov function in the case where $m_r \equiv 1$ or in the case when the system load is small, in which case $(1 - p_f)^m \approx 1 - m_f p_f$. As $t \uparrow$, $n_r(t) \to \nu_r/\mu_r$, hence for by considering sufficiently large t,

we can show that the function

$$\mathcal{U}(x,n) = \sum_{r\in R} \frac{\nu_r}{\mu_r} \log x_r + \sum_{f\in F} \frac{1}{m_f \lambda_f} \left(\nu_f n_f - \mu_f \frac{{n_f}^2}{2} \right) - \sum_{j\in J} \int^{\sum_{r\ni j} \frac{\nu_r}{\mu_r} x_r + \sum_{f\ni j} n_f(t)\lambda_f} p_j(y)dy \tag{27}$$

is a Lyapunov function for the system. Hence with mild constraints on p, all trajectories converge to the unique fixed point.

Notice at this fixed point,

$$p_j = p_j \left(\sum_{r\ni j} \frac{\nu_r}{\mu_r} \frac{w_r}{p_r} + \sum_{f\ni j} \frac{\nu_f}{\mu_f} \left(1 - p_f\right)\right)^{m_f} \lambda_f \Bigg) \tag{28}$$

thus given the load upon the network, we can calculate the appropriate equilibrium rejection probabilities for type f calls, namely $(1-p_f)^{m_f}$.

8 Guarantees

For adaptive traffic, traffic streams are allocated resource in proportion to what they are prepared to pay. So an end-system prepared to pay twice as much as another user, e.g. $w_1 = 2w_2$, receives twice as much. (We can without loss of generality assume all uses are willingness-to-pay users, since more general users can be thought of has having load dependent w_r — see Section2). We can go further than this if we bound the total demand of the network: consider the example of the last section, but where there is no fixed traffic (i.e. $\nu_f \equiv 0$). First, notice that

$$x_r = w_r/p_r \tag{29}$$

hence user r has a throughput of at least w_r provided the system 'price-matched', that is provided $p_r < 1$ for all r. To expand this: suppose each resource is a form of buffered resource, which can serve packets at maximum rate c_j, hence a resource will be overloaded if the load exceeds c_j. Then a sufficient condition for resource j not to be overloaded is

$$\sum_{r\ni j} \frac{\nu_r}{\mu_r} w_r < c_j \tag{30}$$

which says that resource j can mark more packets than the users passing through it are willing to bear. This is a strong condition with $|J|$ constraints, but lays the groundwork for guarantees to be given. Not that the constraint is the *the maximum the users can pay*, rather than the maximum loading on the network. There are corresponding conditions for the users, and corresponding necessary 'max-flow' type bounds: if a flow denotes the vector of 2-tuples n_r, x_r, then we require

$$\sum_{r:x_r>0} n_r w_r < \sum_{j:y_j>0} c_j \tag{31}$$

In practice, for a 'properly-sized' network, we would like the probability of marking to be reasonably small, $p_r < p_{max}$, say not more than 10% on average across routes. In this case, user r will achieve a throughput of a least w_r/p_{max}. Necessary conditions then are that

$$\sum_{r:x_r>0} \frac{1}{p_{max}} n_r w_r < \sum_{j:y_j>0} c_j \tag{32}$$

or stricter conditions are given by

$$\sum_{j\in r} p_j \left(\sum_{r\ni j} \frac{\nu_r}{\mu_r} \frac{w_r}{p_{max}} \right) < p_{max} \quad \forall r \ . \tag{33}$$

Analogous equations hold in the case that $\nu_f > 0$: a sufficient condition for stability is

$$\sum_{r\ni j} \frac{\nu_r}{\mu_r} w_r + \sum_{f\ni j} \frac{\nu_f}{\mu_f} \lambda_f < c_j \tag{34}$$

while if we want the probability of marking to be less than p_{max}, which is a lower bound on the probability of *rejecting* a type f call, then necessary conditions are that

$$\sum_{j \in r} p_j \left(\sum_{r \ni j} \frac{\nu_r}{\mu_r} \frac{w_r}{p_{max}} + \sum_{f \ni j} \frac{\nu_f}{\mu_f} \lambda_f \left(1 - p_{max}\right)^{m_f} \right) < p_{max} \quad \forall r. \tag{35}$$

The use of the above equations mean that is possible to give 'hard' or 'soft' guarantees provided we constrain or bound the user's ability to respond to marks appropriately.

9 Concluding Remarks

We have described a general framework that builds a differentiated services model from simple components: resources in a network (e.g. routers) mark packets when they are congested and the end-systems react to these marks. The end-systems can react as they please provided they have an incentive to react to marks, that is marked packets count as a cost to the user. This may represent real money, but initially at least is more likely to be some form of distributed mint or credit system. For example, we could bound the rate at which users or a particular application can spend consume marks. By bounding the aggregate rate offered to the network, the last section showed how it is possible to give guarantees on quality of service. An edge device may constrain the aggregate marking for a group of systems, in which case it is possible to trade-off marks between applications or end-systems in the same aggregate [10]. For example, in a multimedia application we may choose to preserve the audio quality at the expense of the video, by passing the audio's marks to the video stream.

In optimisation terms (see Section 2) we are solving the Primal problem, where the users average information from the resources (with averaging parameter κ_r), but the resources send back instantaneous information. A complementary approach is used by Low et al [14], who use a Primal-Dual approach where the resources also use an updating function, much as RED uses an average of the queue length to mark packets rather than the instantaneous value. It is interesting to note that for the user, the optimal gain parameter depends on T_r, the reciprocal of the round-trip time, which can be found by the user; one could conjecture that it may be hard for a resource to correctly average information for sources which have very different round trips times, suggesting that averaging at a resource may be difficult and even slow convergence.

If packet marking can be accomplished with a single bit, then the current ECN RFC provides a suitable mechanism for marking packets. The RFC is targeted specifically at TCP: we suggest broadening its scope so that ECN is seen as an IP level mark, and passed up the network stack so that all applications can potentially react to the marks, and not just those based on TCP. Moreover, we wish to allow a variety of reactions to marks, rather than just TCP behaviour.

It is an open question as to whether a single bit is sufficient. Strictly speaking, we should add marks across resources, however little is lost by taking the maximum of the marks along a route rather than the sum provided that the network is lightly loaded (congestion is low). There are other reasons why multiple bits may be useful: for example for identifying where congestion occurs as a packet traverses different domains, or perhaps to communicate an average rate to flows just starting, or when the number of flows is small. Barham and Stratford [23] describe an experimental implementation built on top of Windows2000 which uses a 3rd party traffic controller to alter the rates of the Windows2000 traffic shapers in response to feedback signals, and uses multiple bit marking.

References

1. Robert Axelrod. *The Evolution of Cooperation.* Basic Books, NY, 1984.
2. T. Bonald and L. Massoulié. Impact of fairness on Internet performance. Submitted for publication, 2000.
3. S. Floyd and V. Jacobson. Random Early Detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.
   http://www-nrg.ee.lbl.gov/floyd/red.html.
4. R. J. Gibbens and F. P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35:1969–1985, 1999.
   http://www.statslab.cam.ac.uk/∼frank/PAPERS/evol.html.
5. R. J. Gibbens and P. B. Key. The use of games to assess user strategies for differential quality of service in the internet. In *Workshop on Internet Service Quality Economics*, MIT, December 1999.
   http://research.microsoft.com/research/network/publications/gibkey1999.

6. F. P. Kelly, P. B. Key, and S. Zachary. Distributed admission control. *IEEE Journal on Selected Areas in Communications*, 2000.
http://research.microsoft.com/research/network/publications/dac.htm.
7. F. P. Kelly, A. K. Maulloo, and D. K. H Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.
8. P. B. Key and L. Massoulié. User policies in a network implementing congestion pricing. In *Workshop on Internet Service Quality Economics*, MIT, December 1999.
http://research.microsoft.com/research/network/publications/gibkey1999.ps.
9. P. B. Key and D. R. McAuley. Differential QoS and pricing in networks: where flow control meets game theory. *IEE Proceedings Software*, 146(2):39–43, March 1999.
10. Peter Key, Derek McAuley, Paul Barham, and Koenraad Laevens. Congestion pricing for congestion avoidance. Microsoft Research Technical Report MSR-TR-99-15, MSR, 1999.
http://research.microsoft.com/pubs/.
11. S. Kunniyur and R. Srikant. End-to-end congestion control schemes: Utility functions, random losses and ECN marks. In *INFOCOM 2000*, 2000.
12. Koenraad Laevens, Peter Key, Derek McAuley, and Paul Barham. An ecn-based end-to-end congestion-control framework: experiments and evaluation. Microsoft Research Technical Report MSR-TR-2000-104, MSR, 2000.
http://research.microsoft.com/research/network/publications/MSRTR2000_104.pdf.
13. D. E. Lapsley and S. H. Low. Random Early Marking: an optimisation approach to Internet congestion control. In *Proceedings of IEEE ICON'99*. IEEE, 1999. Brisbane, Australia.
14. S. H. Low and D.E. Lapsley. Optimization flow control – I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, dec 1999.
15. J. K. MacKie-Mason and H. R. Varian. Pricing congestible network resources. *IEEE Journal of Selected Areas in Communications*, 13(7):1141–1149, 1995.
16. L. Massoulié. Stability of distributed congestion control with heterogeneous feedback delays. Tech Report MSR-2000-111, Microsoft Research, 2000.
17. Laurent Massoulié, Peter B. Key, and Koenraad Laevens. End-user policies for predicting congestion patterns in data networks. In *13th ITC Specialist Seminar on Internet Traffic Measurement*, September 2000.
http://research.microsoft.com/research/network/publications/monterey.ps.
18. R. Mazumbdar, L.G. Mason, and C. Douglieris. Fairness in network optimal control: optimality of product forms. *IEEE Trans. Communications*, 39:775–782, 1991.
19. J. Mo and J. Walrand. Fair end-to-end window based congestion control. In *SPIE 98, International Symposium on Voice, Video and Data Communications*, 1998.
20. John F Nash. The bargaining problem. *Econometrica*, 1950.
21. K. Ramakrishnan and S. Floyd. A proposal to add explicit congestion notification (ECN) to IP. RFC 2481, IETF, january 1999. ftp://ftp.isi.edu/in-notes/rfc2481.txt.
22. S. Shenker. Fundamental design issues for the future Internet. *IEEE J. Selected Area Communications*, 13:1176–1188, 1995.
23. Neil Stratford. Congestion pricing: A testbed implementation. In *Multi-Service Networks 2000*, July 2000.
http://research.microsoft.com/research/network/talks/Neil_S_Cos2k.pdf.
24. D. K. H. Tan. *Mathematical models of rate control for communication networks*. PhD thesis, University of Cambridge, 1999.
http://www.statslab.cam.ac.uk/ dkht2/phd.html.
25. Damon Wischik. How to mark fairly. In *Workshop on Internet Service Quality Economics*. MIT, Dec 1999.

Appendix: Fairness and Mandated Controls

We can generalise the approach of Section 2 by letting the Network mandate that users should adjust their rates according to the update rule

$$\frac{d}{dt}x_r(t) = \kappa_r \left(x_r\left(t\right) F_r'\left(x_r\left(t\right)\right) - x_r\left(t\right) \sum_{j \in r} p_j\left(y_j\left(t\right)\right) \right) \tag{36}$$

where F_r is a function of the form

$$F_r(x) = w_r \frac{x^{1-\alpha}}{1-\alpha} \tag{37}$$

for some fixed $\alpha \neq 1$. The Network implicitly maximises $\sum_r F_r(x_r) - \sum_j C_j(y_j)$, and in the case that $w_r \equiv 1$ the resulting allocation will correspond to a Maximum Utilisation if $\alpha = 0$, corresponds to Proportional Fairness if $\alpha \to 1$ and Max-Min fairness as $\alpha \uparrow$ [19], and weighted variants of these for general w_r. At the optimum,

$$F_r'(x) \equiv w_r {x_r}^{-\alpha} = p_r \tag{38}$$

If the user adapts the parameter w_r over time according to

$$w_r = {x_r}^{\alpha} U_r'(x_r) \tag{39}$$

or equivalently is seeking to solve

$$\text{Maximise} \quad U_r\left(\left(\frac{w_r}{p_r}\right)^{\frac{1}{\alpha}}\right) - p_r\left(\frac{w_r}{p_r}\right)^{\frac{1}{\alpha}} \quad \text{over} \quad w_r \geq 0. \tag{40}$$

then we can show that the User Optimum and Social optimum again coincide, based upon an underlying 'α-Fairness' model.

In this case w_r has the interpretation that $w_r {x_r}^{1-\alpha}$ is the amount the user is prepared to pay per unit time. The case $\alpha = 1$ is equivalent to $F_r(x_r) = w_r \log x_r$